IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
Bruce E. Lavigne, Paul T. Congdon,
and Mark Gooch

Examiner: Hossain, Ibrahim M.

Serial No. 10/723,041

Art Unit: 2145

Filing Date: November 26, 2003

Attorney Docket No.: 200311029-1

Title: Remote Mirroring Using IP Encapsulation

Declaration by Inventors under 37 C.F.R. 1.131

1. The persons making this declaration are the inventors of the above-referenced patent application.

2. The following attached documents are submitted as evidence.

Exhibit A: Copy of PowerPoint presentation, dated January 23, 2002.
Exhibit B: Copy of Bttf Mirroring Proposal, dated October 1, 2002.
Exhibit C: Copy of HP Invention Disclosure PDNO 200310880-1, dated March 24, 2003.

3. From Exhibit A, it can be seen that conception of the invention in the above-referenced application was made by at least January 23, 2002.

4. The following facts are submitted to establish the diligence of the applicants from at least January 23, 2002 until the filing of the above-referenced patent application.

a. A proposal was developed for reducing the invention to practice in an actual product. The proposal is shown in Exhibit B, dated October 1, 2002.

b. An invention disclosure was prepared for pursuing patent protection on the invention. The invention disclosure is shown in Exhibit C, dated March 24, 2003.

c. The above-referenced patent application was filed on November 26, 2003.

5. As a person signing below:

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Bruce E. Lavigne
Bruce E. Lavigne

7/30/07
Date

Paul T. Congdon
Paul T. Congdon

8/6/2007
Date

Mark Gooch
Mark Gooch

30th JULY 2007
Date

Attachments: Exhibits A, B and C

Exhibit A

Remote Port Monitoring

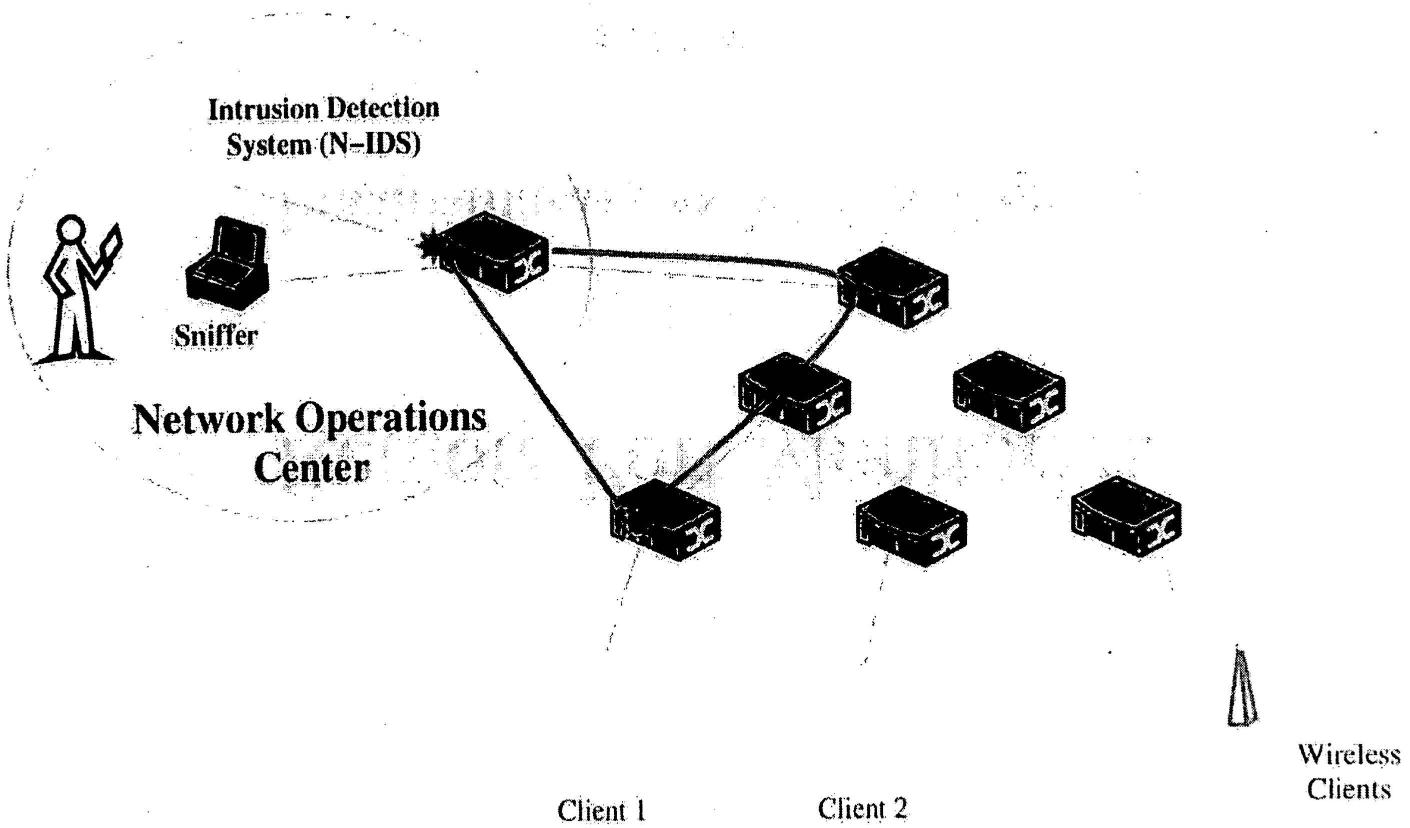
Encapsulation vs VLAN Tagging

Paul Congdon
01/23/02

Draft

What is Remote Port Monitoring

Directing 'Port Monitor' Traffic To A Distant Switch



Current Port Mirroring

- Designed to function on a standalone switch only. Typically requires a sniffer or IDS to be attached to the switch directly.
- Operation varies by switch, but primarily involves the following:
  - * Select a monitor port (the port where monitor traffic will be sent out)
  - * Select a port, set of ports or VLAN whose traffic should be monitored
  - * All traffic to and from the selected ports is sent to the monitor port as seen on those ports.

Advantages of Remote Monitoring

- Analysis equipment can be expensive and hard to manage * Desire to keep it in the NOC
  - Cisco and Enterasys IDS systems are $$$
  - The more high–powered the server the better the IDS will perform
  - Sniffers are expensive and difficult to use
- Physical access to wiring closets and wireless workgroups can be difficult
- Monitoring multiple low–bandwidth streams at once is valuable (e.g. correlation of activity at wireless access points).
- Enables on–demand centralized auditing and tracking facilities.

Disadvantages of Remote Monitoring

- May be difficult to configure
- Requires additional backbone resources and bandwidth
- Potential for congestion or dropped packets along the way may provide an incomplete picture
- Is it just the Lunatic Fringe that cares?

Some Ways to Get Monitor Traffic Back to the NOC

- Wire a separate network from each switch
  - ✓ Create an overlay network with dedicated links.
- Encapsulates and forward
  - ✓ Add a header to the front of monitored packets and forward out a shared link.
  - ✓ A destination switch will remove header and forward out local monitor port in original format.
- Copy and VLAN tag
  - ✓ Monitored packets are VLAN tagged (or re–tagged) and forwarded out a shared link.
  - ✓ Last hop switch removes VLAN tag and forwards out local monitor port in untagged format.

Wiring A Separate Network

* Logistically difficult

* Expensive. Little or no resource sharing

* Can' t de–multiplex multiple monitored port' s traffic

* Any shared internal switches must be an MFD VLAN switches

* Unclear how separate network handles topology changes (e.g. Spanning Tree)

\+ Dedicated bandwidth

\+ Can be done today. No changes required

Pro = '+', Con = '–'

Encapsulate Monitored Packets Inside a New Packet

* Complex operations for switches (both encaps and decaps)
* Creates jumbo frames
* Shares backbone bandwidth with regular traffic
\+ Transparently support 3rd party backbone switches (jumbos required)
\+ Capable of creating multiple virtual monitor streams
\+ Traffic follows normal forwarding rules and supports topology changes

Pro = '+', Con = '–'

Copy and VLAN Tag Monitored Packets onto a New VLAN

* Works for untagged sources only. Loose any original VLAN tag information.

* Requires consistent end–to–end configuration

* Burns a VLAN for each unique monitor stream

* All shared switches along the path must be MFD VLAN switches

* Unclear how intermediate switches would adjust to changes in topology

\+ Switches already do VLAN tagging, so minor complexity addition

Pro = '+', Con = '–'

Ideas on Tagging

- Configuration
  - Similar to current scheme, but instead of defining only a monitor port, a destination VLAN and egress format (tagged or untagged) is also defined. Call the combination of monitor port, VLAN and format a monitor stream.
  - Multiple monitor streams may be defined on a single switch.
  - The monitor port of the monitor stream may be shared with other normal traffic and other monitor streams.
  - Ports and/or VLANs to be monitored are assigned to one of the monitor streams. Similar to how they are assigned to a monitor port today.
  - Intermediate switches could be configured to monitor the same VLAN used in the monitor stream, resulting in simple stream forwarding.
- Implementation Considerations
  - Monitored frames must be copied and tagged separate from normal forwarding. Somewhat like Multicast routing today.
  - It should be possible to re–tag a frame with the same VID if necessary (stream forwarding). Alternatively, use existing VLAN monitoring scheme on intermediate switches.

More Ideas on Tagging

- More Implementation Considerations
  - It might be possible to simply the configuration by defining a special VLAN as the mirroring VLAN and intermediate switches would use special forwarding rules for this VLAN.
  - It must be possible to disable hardware learning on this special VLAN and it must be possible to limit the list of flood ports for this VLAN as well.
  - The egress switch could send an out–of–band multicast packet (periodically) to the ingress switch. The path learned for this multicast would be the path used to forward traffic for the special VLAN.
  - Flooding for the mirroring VLAN would only go out the port learned from the above multicast packet. If the topology changes, you would only flood the backbone until the path was learned again.
  - There would be no entries in the address table for this VLAN so all traffic would be flooded, but the flooding would be controlled as described above.
- Configuration in this scheme
  - Ingress and Egress switches would have to know they were remote end–points. The ingress needs to know so it can tag and copy appropriately. The egress needs to know so it can generate multicast path packets
  - The intermediate switches need to have the special VLAN configured as tagged on all ports (by default) and restricted by a user to more ports if desired. This special VLAN would have the special forwarding rules implemented as described above.

Questions with Tagging

- How can intermediate switches be automatically programmed using something like GVRP?
- How would changes in the active topology be handled (e.g. Spanning Tree reconfiguration)?
- How does this work with x–RMON in the switch and in the network as a whole?

Ideas on Encapsulation

- Configuration
  - Should only require configuration on edge switches (encapsulation) and destination switch (de–capsulation).
  - Must associate and define a L2/L3 header for a monitor stream. Should map it to a VLAN as well.
  - Not necessary to specify a monitor port for forwarding, but may be desirable (see specific egress port forwarding below).
  - Ports and/or VLANs to be monitored are assigned to one of the monitor streams. Similar to how they are assigned to a monitor port today.
  - On the switch where streams are directed, a de–capsulation port must be configured for each monitor stream. Address information in the header should direct traffic to this switch.
- Implementation Considerations
  - To use normal forwarding on source switch, copy and encapsulation must be done on ingress.
  - To use specific egress port forwarding on source switch, a header block of data can be appended to the monitored frame. Special internal transfer information should accompany the frame for encapsulation instructions and to include dynamic data in the header (e.g. source port)
  - Encapsulation header should include addressing, VLAN, ethertype/protocol numbers to trigger decapsulator, encaps and decaps port numbers, other,...
  - Out–of–band control protocol may be implemented to assure the destination is available and ready to decaps a monitor stream. Some necessary configuration information could be passed in this protocol to avoid manual entry on both switches.

More on Encapsulation

- Options
  - * Should allow a mode where packets are truncated as well as encapsulated so as to not create jumbos
  - * Should allow an egress mode that preserves the encapsulation header to allow scheme–aware IDS systems to be built
- Header fields to consider
  - * Ethernet encapsulation of an IP/UDP packet, thus include MAC SA/DA, IP SA/DA, UDP protocol number as programmable fields. Set the don' t fragment bit in the IP header. Decaps switch must be able to easily decode this packet.
  - * Note: Need to allocate a UDP protocol number for this use
  - * Allow for source port to be filled–in by ingress switch
  - * Allow for a sequence number that is associated with the source port and if possible, all packets of the egress stream. This allows decaps switch to know if packets have been dropped somewhere along the way
  - * Consider including egress port number and egress format for the decaps switch in the packet as well to simplify the decaps process. Consider carrying any fields that help the decaps switch and can be set–up by the software out–of–band stream initialization protocol.
  - * Consider including some kind of relative timestamp field as well to understand jitter.

Questions on Encapsulation

- Is it easier to perform the encapsulation on ingress or egress?
- Any security concerns worth considering?
- How might this impact x–RMON in the switch or in the network in general?

Exhibit B

BttF Remote Mirroring Proposal

Hewlett-Packard, Procurve Networking Business

HEWLETT PACKARD
ProCurve Networking

Contact: Bruce LaVigne
bruce@rose.hp.com

Revision	*: 1.0*
Revision Date	*: October 1, 2002 5:53 pm*

(This page intentionally left blank)

Mirroring Overview 1

The concept behind mirroring is to replicate packets that someone is interested in to somewhere else where they can be captured and/or examined. Packets that someone is interested in may be to/from a particular physical port (port mirroring), to/from a particular VLAN (VLAN mirroring), or based on other criteria (ACL mirroring, BMP mirroring, Hash entry mirroring). Replication may be local or remote, but a key feature is to **exactly** replicate the mirrored packets on to the capturing/examining equipment, including VLAN tags, TOS, TTL, etc. In addition, sometimes it is useful to manage the capturing/examining equipment over the same ethernet link, and other times it is desirable to only have mirrored traffic appear on the link to the capturing/examining equipment.

This proposal extends the basic mirroring found in Alpha3 today by adding additional mirroring sources and additional (remote) mirroring destinations. This not only brings us up to the rest of the industry, but surpasses it -- one more BttF differentiator!

1.1 Terminology

The terms used both internally and industry-wide to describe mirroring today are very confusing and often ambiguous (mirror, monitor, snoop, probe, sniffer, SPAN, Roving Analysis, etc.), so we will define the terms used in this proposal so at least it is clear what we are talking about here. These terms may or may not end up in general usage or visible to the customer. See the listed relevant subsection for a more thorough description.

Mirroring	The overall feature of replicating packets from some selected source(s) to a selected destination.
Mirror Session	One instance of mirroring configured on the switch. We propose a total of 4 possible sessions for BttF, each fully configurable and separate from each other. See section 4.1 for more details. Many vendors support multiple mirror ports. Cisco currently supports between 1-4 sessions on various product lines, and uses the same term, session, as well.
Local Mirroring	The process of copying packets from enabled sources to a single configured port local to this switch. This is the only form of mirroring present on Alpha3, and on most networking gear. See section 3.1 for more details. Cisco refers to this as Switched Port ANalyzer (SPAN) or Port Snooping.

VLAN Remote Mirroring	The process of copying packets from enabled sources to a configured VLAN on this switch. Generally, the VLAN will be propagated to other switches which are members of this VLAN. See section 3.2 for more details. Cisco is the only other vendor who has this (that I could find), and refers to this as Remote SPAN (RSPAN).
MAC Encapsulated Remote Mirroring	The process of copying and encapsulating packets from enabled sources to a configured destination MAC/VLAN/encap/L3 protocol on this switch. Generally, the final destination of the destination MAC address is some other switch, allowing remote mirroring across a L2 domain while maintaining the original VLAN tag and the ability to pass through older networking gear. See section 3.3 for more details. No other networking vendor has anything like this to my knowledge.
IPv4 Encapsulated Remote Mirroring	The process of copying and encapsulating packets from enabled sources to a configured destination IP/MAC/VLAN/L4 protocol on this switch. Generally, the final destination of the destination IPv4 address is some other switch, allowing remote mirroring across a L3 domain while maintaining the original MAC/IP/VLAN tag and the ability to pass through older switching/routing gear. See section 3.4 for more details. No other networking vendor has anything like this to my knowledge.
Extended Mirroring	The process of copying packets from enabled sources in an extended form (includes ISH) to a configured destination. This allows us to send packets to a Special Function Card based on things like output port, which the current copy logic does not permit, or to send extended packets to a (possibly remote) sniffer during debug. No other networking vendor has anything like this to my knowledge.
Mirror Source Port (MSP)	A port whose ingress (rx) or egress (tx) traffic (or both) is copied from in the mirroring operation. This is referred to as Port-based mirroring (Cisco's PSPAN). See section 2.1 for more details.
Mirror Source VLAN (MSV)	A VLAN whose ingress (rx) or egress (tx) traffic (or both) is copied from in the mirroring operation. See section 2.2 for more details. This is referred to as VLAN-based mirroring (Cisco's VSPAN).

Mirror Source MAC (MSM)	A MAC address whose ingress (rx) or egress (tx) traffic (or both) is copied from in the mirroring operation. This is referred to as MAC-based mirroring (it comes from the L2 hash table). See section 2.3 for more details. No other networking vendor has a counterpart that I know of.
Mirror Source IP (MSI)	An IP address whose ingress (rx) or egress (tx) traffic (or both) is copied from in the mirroring operation. This is referred to as IP-based mirroring (it comes from the L3 hash table). Note that this would apply to routed packets only. See section 2.4 for more details. No other networking vendor has a counterpart that I know of.
Mirror Source Subnet (MSS)	An IPv4 subnet address whose egress (tx) traffic is copied from in the mirroring operation. This is referred to as Subnet-based mirroring (it comes from the BMP table - and thus has no ingress variant). See section 2.5 for more details. No other networking vendor has a counterpart that I know of.
Mirror Source ACL (MSA)	An ACL match whose traffic is copied from in the mirroring operation. This is referred to as ACL-based mirroring (it comes from the ACL table - and thus is effectively both ingress and egress, depending on the ACL fields selected). See section 2.6 for more details. No other networking vendor has a counterpart that I know of.
Mirror Dest Port (MDP)	The port defined as the destination for Local Mirroring. All packets from the various enabled Mirror Source locations will be copied to this port. See section 3.1 for more details.
Mirror Dest VLAN (MDV)	The VLAN defined as the destination for VLAN Remote Mirroring. All packets from the various enabled Mirror Source locations will be copied to this VLAN. See section 3.2 for more details.
Mirror Dest MAC (MDM)	The MAC address encapsulation defined as the destination for MAC Encapsulated Remote Mirroring. All packets from the various enabled Mirror Source locations will be copied to this MAC address by encapsulating them with either an 18-byte Ethernet II header consisting of DA/SA/VLAN/ETYPE/RESERVED/FLAGS/TCI, or a 36-byte SNAP

header consisting of DA/SA/VLAN/LEN/LSAP/CTL/ORG/ETYPE/RESERVED/FLAGS/TCI. See section 3.3 for more details.

Mirror Dest IP (MDI)

The IPv4 address encapsulation defined as the destination for IP Encapsulated Remote Mirroring. All packets from the various enabled Mirror Source locations will be copied to this IPv4 address by encapsulating them in a 36-byte header consisting of DA/SA/VLAN/IPv4_HDR. See section 3.4 for more details.

Mirror Truncate Length

The maximum internal size of the mirrored packet. Two reasons to truncate are: 1) to prevent oversize packets in the network when using Mirror Encapsulation, and 2) to lessen the network load by only mirroring the first portion of packets. This may be set to zero to never truncate. Since this is an internal size, it is configured in 18-byte blocks. The conversion from internal length to actual packet length is described in the *BttF Architecture Interchip Interface Specification*. See section 4.2 for more details.

Mirror Exit Switch (MES)

The switch which is the exit point for a MAC or IP encapsulated remote mirror stream. This is determined by the DA lookup, which returns a "mirror_exit" bit in the ISH. At this point, the packet is decapsulated in hardware and sent out the destination port exactly as it appeared on entry to the mirror "tunnel".

Mirror Sources 2

There are six types of sources for mirrored packets. Note that some types can have ingress (rx) traffic enabled, some egress (tx) traffic enabled, and some both enabled. Also note that in the current definition of BttF, some types are not implementable since we are short on bits in some lookup tables. In general, if more than one type of source is enabled for a particular mirror session, then traffic matching any of the configured types is mirrored (see the description in port and VLAN mirroring for the one exception to this rule).

2.1 Port-based Mirroring

Port-based mirroring allows ingress (rx) and/or egress (tx) traffic on a port (network or CPU) to be copied to the mirror destination. Any number of source ports may be specified per mirror session, in any combination of rx, tx, or both.

Note – The combination of port-based mirroring with VLAN-based mirroring below may be configured as an OR operation (as in current Alpha3), or as an AND operation, where in order to mirror a packet, it must belong both to a MSP and a MSV. This matches Cisco's "filter" option.

2.2 VLAN-based Mirroring

VLAN-based mirroring allows ingress (rx) and/or egress (tx) traffic on a VLAN to be copied to the mirror destination. One VLAN may be mirrored per mirror session, with either rx, tx, or both enabled.

Note – The combination of port-based mirroring above with VLAN-based mirroring may be configured as an OR operation (as in current Alpha3), or as an AND operation, where in order to mirror a packet, it must belong both to a MSP and a MSV. This matches Cisco's "filter" option.

2.3 MAC-based Mirroring

MAC-based mirroring allows ingress (rx) and/or egress (tx) traffic matching a MAC-VID pair (an entry in the MAC hash table) to be copied to the mirror destination. Any number of hash entries may be programmed per mirror session, in any combination of rx, tx, or both.

Implementation Note – Current BttF MAC hash entry definition has two bits set aside for this function - one used on the source lookup, one used on the destination lookup. Each bit has a mapping to one of the four mirror sessions, configured per chip in the HPP. Thus, we support any one mirror session for source and any one mirror session for destination, with any number of hash entries able to set these mirror session bits.

Implementation Note - If the BttF box has routing enabled, software should scan through the L3 hash and BMP tables for the egress MAC address in question as well, since we don't do another dest MAC lookup once we've done a dest IP lookup. If it finds the egress MAC, it can set the mirror bit in those tables also.

2.4 IP-based Mirroring

IP-based mirroring allows ingress (rx) and/or egress (tx) traffic matching an IP-VID pair (SA, flows) or IP address (DA) (an entry in the IP hash table) to be copied to the mirror destination. Any number of hash entries may be programmed per mirror session, in any combination of rx, tx, or both.

Implementation Note - Current BttF L3 hash lookup is only performed on routed packets, so bridged IP packets will not hit on these entries, and will not be selected with this type of mirroring. Also note that the source IP lookup is not done unless security or some other feature is enabled, and the destination IP lookup is not done if the BMP lookup hit and was not listed as having a more specific route.

Implementation Note - Current BttF IP hash entry definition has two bits set aside for this function - one used on the source lookup, one used on the destination lookup. Each bit has a mapping to one of the four mirror sessions, configured per chip in the HPP. Thus, we support any one mirror session for source and any one mirror session for destination, with any number of hash entries able to set these mirror session bits.

2.5 Subnet-based Mirroring

Subnet-based mirroring allows egress (tx) traffic matching an IP subnet address (an entry in the BMP table) to be copied to the mirror destination. Any number of table entries may be programmed per mirror session.

Implementation Note - Current BttF BMP table entry definition has a single bit set aside for this function, with a mapping to one of the four mirror sessions configured per chip in the HPP. Thus, we support any one mirror session, with any number of BMP entries able to set the mirror session bit.

2.6 ACL-based Mirroring

ACL-based mirroring allows traffic matching an ACL entry (ternary match on IP_DA, IP_SA, TCP/UDP source & destination port numbers, TCP Syn & Ack bits, & physical ingress port) to be copied to the mirror destination. Any number of table entries may be programmed per mirror session. Since both source (ingress) and destination (egress) IP addresses are available in the ternary CAM, we can support rx, tx, or both on each entry. This form of mirroring is more powerful than the above two forms, in that with the ternary CAM, we can perform IP-based mirroring, Subnet-based mirroring, or flow-based mirroring, and additionally match on fields not present in any of the other forms. Also, these lookups are done for both bridged and routed IP packets. The downside is table space -- first off, there are only 1.5K CAM entries today, and secondly, configuring mirroring with other ACLs in the system will require some form of ACL compiler or operator intervention.

Mirror Destinations 3

There are four types of destinations for mirrored packets. The destination types are mutually exclusive -- only one type of destination may be configured at a time on each mirror session. Any destination type may be configured as normal or extended, lossy or lossless, throttlable or not, forced tagged/untagged, and can override out_queue.

3.1 Local Mirroring

Local mirroring sends the mirrored traffic to a specific port (the Mirror Destination Port, MDP) on this switch. This is the most basic form of mirroring, and is the only form of mirroring present on Alpha3 switches.

Note – Following Cisco's lead, we probably want to allow software to optionally disable learning, and optionally not forward ingress traffic from the MDP. For BttF, this can be accomplished by setting the port MIST state to "not_a_member".

Implementation Note – Packet tagging my not exactly match the source for Local Mirroring, due to port configuration differences (PPVID registers). If this is an issue, one must use one of the more advanced Encapsulated Remote Mirroring methods.

Implementation Note – The MDP may not be a member of a mesh, for this will break the meshing code.

3.2 VLAN Remote Mirroring

VLAN remote mirroring floods the mirrored traffic to a specific VLAN (the Mirror Destination VLAN, MDV) on this switch. This VLAN can be configured on multiple switches to propagate (flood) mirrored traffic in a layer 2 domain to a sniffer on another switch. Since the mirrored packets are copied to the MDV, the original VLAN tag is lost with this type of mirroring. This is how Cisco's RSPAN works.

Note – Again, we probably want to optionally disable learning on this VLAN. For BttF, this can be accomplished by setting the VID as illegal.

Implementation Note – The MDV may not contain any ports which are a member of a mesh, for this will break the meshing code.

3.3 MAC Encapsulated Remote Mirroring

MAC Encapsulated remote mirroring encapsulates the mirrored traffic by adding either an 18-byte Ethernet II header or a 36-byte SNAP header to the beginning of every packet, and forwarding the resulting packet out using a software configured logical port. This allows mirrored packets to traverse any number of switches (belonging to any vendor) in a layer 2 domain before being decapsulated in hardware at the destination switch (which must be a BttF product appropriately configured) and forwarded out a sniffer port. The format of the 18-byte Ethernet II header is:

destination_mac_addr	source_mac_addr	etype	rsvd	flags	TCI

Figure 3-1 MAC Ethernet II Encapsulated Remote Mirroring Header

The format of the 36-byte SNAP header is:

destination_mac_addr	source_mac_addr	Length	LSAP	CTL	ORG
ORG (cont)	etype	reserved		flags	TCI

Figure 3-2 MAC SNAP Encapsulated Remote Mirroring Header

Software writes all 18 or 36 bytes of this header, along with a logical port register and some override bits (force tag/untag, extended, out_queue & override, throttlable, reclass_to, lossless).

During packet encapsulation, the 16-bit TCI field (COS/TR/VID) which software writes is moved into the SEND header, where it becomes part of the outgoing VLAN tag (if tagged). The mirrored packet's TCI field is moved into the header field, so that on decapsulation, it can be properly placed in the restored packet's VLAN tag. The upper bit of the flags field (0x80) is written by the ASIC into the header indicating whether the restored packet should be tagged or not. This operation essentially preserves the original VLAN tag of the packet so we exactly replicate the packet to the sniffer port. The rest of the flags field is currently reserved for use by the ASIC. The determination of whether the restored packet should be tagged is as follows: for input, if the packet arrived tagged, it should be tagged; for output, there will be one set of PPVID registers which software can program in the same way as an output port -- if the packet was egress mirrored and it matches one of the PPVID registers, it will be untagged, otherwise tagged. If there is a conflict (i.e. a packet is mirrored on ingress and egress), tagging wins.

For the SNAP encapsulated packets, the length field is computed based on the (new) xfer_18 value and adj_bytes. If this length is greater than a software configured value, then the length field is not overwritten by the ASIC. This allows software to place the jumbo value 0x8870 in the length portion of the encapsulation, and then decide how large a packet must be before the 0x8870 is used. Setting the length compare to zero forces the software's configured length field on all packets, setting the length compare to a number larger than the max size packet (say, 32K) forces the length to be always computed (even if it generates values which could be construed as an ethertype), and

setting the length to some intermediate value (say, 1518, the default value) uses the software configured length if the packet becomes larger than a normal ethernet frame.

The resulting encapsulated packet is then sent to the configured logical port.

Note that the resulting packet may be larger than the maximum allowed size on some links between the source and the final destination. To address this, we allow software to configure a Mirror Truncate Length (see section 4.2) if it so desires.

It is expected that some higher-layer software -- either via user configuration or via communication with the decapsulating switch -- is determining the correct values of MAC addresses, etype, TCI, etc. to put in the header.

Unresolved Issue - The etype field needs to be chosen with care so as not to conflict with existing protocols. Does HP own one, or are they even available?

Decapsulation is also handled by the ASIC; if the switch detects that it is the Mirror Exit Switch (MES), then the 18-byte (Ethernet II) or 36-byte (SAP/SNAP) encapsulation is stripped, and the VLAN tag is restored from the TCI/flags portion of the encapsulation.

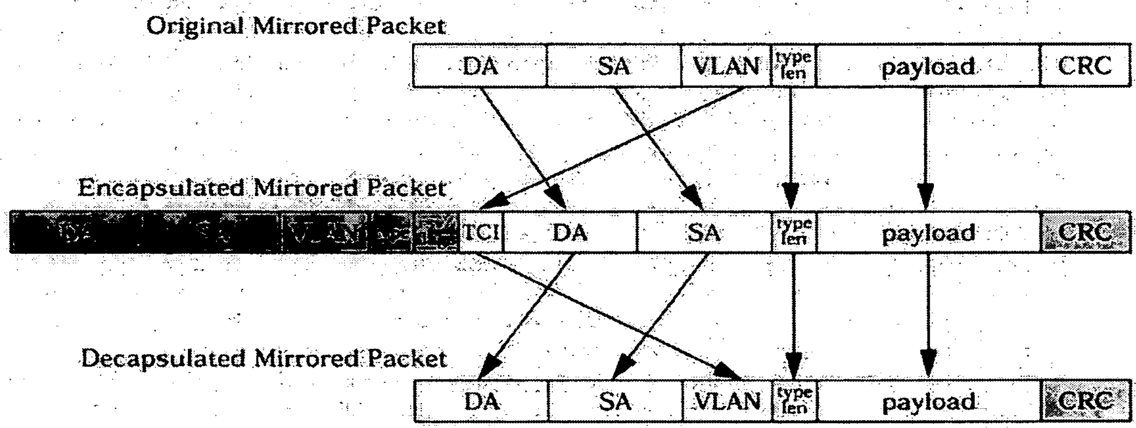


Figure 3-3 MAC Ethernet II Encapsulated Remote Mirroring Packet Formatting

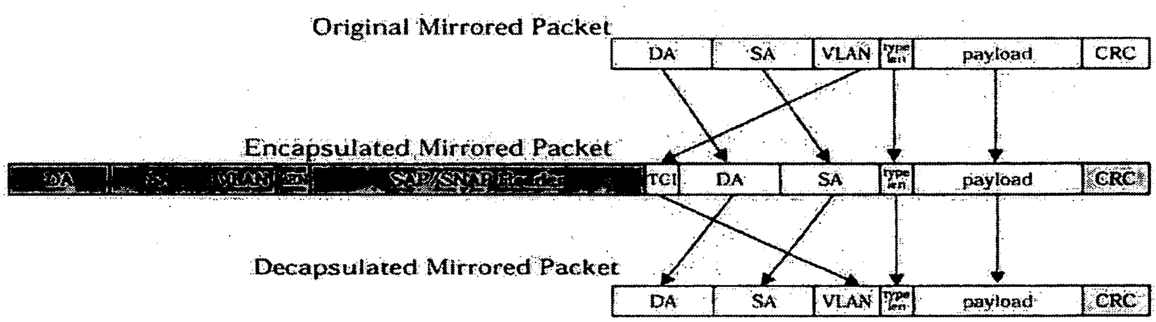


Figure 3-4 MAC SNAP Encapsulated Remote Mirroring Packet Formatting

3.4 IPv4 Encapsulated Remote Mirroring

IPv4 Encapsulated remote mirroring encapsulates the mirrored traffic by adding a 36-byte IPv4 header to the beginning of every packet, and forwarding the resulting packet out using a software configured logical port. This allows mirrored packets to traverse any number of switches or routers (belonging to any vendor) in a layer 3 domain before being decapsulated in hardware and forwarded out a sniffer port. The format of the header is:

<table>
<tr><td colspan="4">destination_mac_addr</td><td colspan="2">source_mac_addr</td><td>0x0800 | 0x45 | TOS</td><td>total length</td></tr>
<tr><td>identif-ication</td><td>flags/ fr_offset</td><td>TTL</td><td>proto</td><td>chksum</td><td>source_ip_addr</td><td>dest_ip_addr</td><td>TCI</td></tr>
</table>

Figure 3-5 IP Encapsulated Remote Mirroring Header

Software writes all 36 bytes of this header, along with a logical port register and some override bits (force tag/untag, extended, out_queue & override, throttlable, reclass_to, lossless).

During packet encapsulation, the 16-bit TCI field (COS/TR/VID) which software writes is moved into the SEND header, where it becomes part of the outgoing VLAN tag. The mirrored packet's TCI field is moved into the header field, so that on de-encapsulation, it can be properly placed in the restored packet's VLAN tag. The upper bit of the identification field (0x8000) is written by the ASIC into the header indicating whether the restored packet should be tagged or not. This operation essentially preserves the original VLAN tag of the packet so we exactly replicate the packet to the sniffer port. The second bit of the identification field (0x4000) is written by the ASIC to indicate which FD is sending this packet, and is used along with the rest of the identification field, written as incrementing numbers by the ASIC, to uniquely identify this packet across the network. Note that software ought to set the "don't fragment" bit -- flags bit 0x02, because the Mirror Exit Switch (MES) will NOT be able to correctly re-assemble fragments and send the original packet to the sniffer. The total length and checksum fields are also properly computed by the ASIC. The determination of whether the restored packet should be tagged is as follows: for input, if the packet arrived tagged, it should be tagged; for output, there will be one set of PPVID registers which software can program in the same way as an output port -- if the packet was egress mirrored and it matches one of the PPVID registers, it will be untagged, otherwise tagged. If there is a conflict (i.e. a packet is mirrored on ingress and egress), tagging wins.

The resulting encapsultated packet is then sent to the configured logical port.

Note that the resulting packet may be larger than the maximum allowed size on some links between the source and the final destination. To address this, we allow software to configure a Mirror Truncate Length (see section 4.2) if it so desires.

It is expected that some higher-layer software -- either via user configuration or via communication with the decapsulating switch -- is determining the correct values of MAC addresses, TCI, IP Addresses, protocol, etc. to put in the header.

Unresolved Issue – The proto field needs to be chosen with care so as not to conflict with existing protocols. Does HP own one, or are they even available?

Unresolved Issue – How does this work in the presence of VPN's or other security features? Do these things need to be disabled when IPv4 remote mirroring?

Note – The next-hop MAC address (da_mac) needs to be updated by software if routing updates cause the mapping from dest-IP to dest-MAC to change.

Decapsulation is also handled by the ASIC; if the switch detects that it is the Mirror Exit Switch (MES), then the 36-byte IPv4 (for hw_rtbl pkts) encapsulation is stripped, and the VLAN tag is restored from the TCI/flags portion of the encapsulation

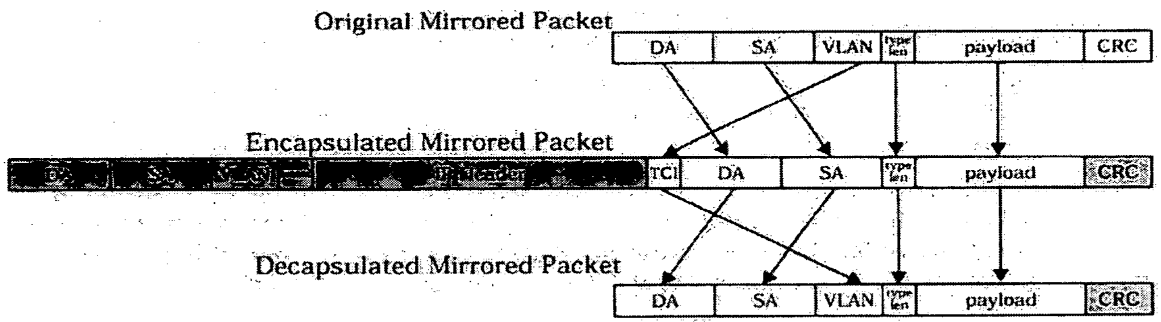


Figure 3-6 IPv4 Encapsulated Remote Mirroring Packet Formatting

Additional Mirroring Features 4

4.1 Mirror Sessions

We propose four individual mirror sessions for BttF. Each is independently configurable on both source(s) and destination.

4.2 Packet Truncation

We propose a set of Mirror Truncate Length registers, one per session. Setting these registers to zero disables truncation. The value of this feature is twofold:

1. Mirrored packets that are encapsulated may grow beyond the maximum transfer size of subsequent links, and thus may be dropped as being oversized unless they are truncated.
2. For performance reasons, it may be beneficial to mirror only the first portion of all packets -- thus, if a full-duplex link was sending line-rate packets with an average packet size of 500 bytes, we could truncate to 250 bytes and mirror both input and output on a link of the same speed without dropping any packets.

4.3 Filtering based on da_type

It may be useful to separately enable/disable mirroring of packets based on da_type, so that we only mirror unicast traffic, or multicast traffic, etc.

4.4 Forcing Best Effort On/Off

Generally, forcing Best Effort mode on mirrored traffic will prevent head-of-line blocking issues -- especially if the mirror link is overloaded with traffic, as would be the case when using a same-speed link to mirror both ingress and egress traffic.

However, it may be desirable, if the user knows the mirrored traffic is light but bursty, to disable Best Effort, and take the risk of head-of-line blocking in order to be assured that all traffic is correctly mirrored to the sniffer.

4.5 Mirroring CPU ports

This should pretty much come for free in the current BttF architecture, but mirroring ingress/egress/both CPU packets may be useful. Probably want a configuration to understand CPU_message packets, to either mirror them or not (customers would NOT want them mirrored, we may want ONLY them mirrored).

Changes/Impacts per Module 5

5.1 Input PreProcessor (IPP)

None

5.2 Hash PreProcessor (HPP)

1. Extra bit in L2 and L3 hash tables - "Mirror Exit", copied to ISH
2. Changes to learn & drop rules to support not learning or forwarding from MDP/MDV
3. Extra "ingress mirror" bits in L2 and L3 hash tables - if there's room
4. Configuration for port/vlan AND vs. OR mirroring

5.3 Chip LookUp Engine (CLUE)

1. Mirror logical port registers, override bits, per session
2. Mirror destination type registers per session
3. Mirror da_type filter registers, per session

5.4 MegaEngine (ME)

None

5.5 Fabric Driver (FD)

1. Mirror Encapsulation RAM, per session
2. Mirror Truncate Length registers, override bits, per session
3. Encapsulation/Decapsulation logic
4. PPVID registers (5) per session to determine tagging
5. Enhanced BEA logic (falls out of reclass_to updates)

5.6 Fabric Receiver (FR)

1. Pass back "sent to MDP" bits in Buffer Reply
2. Configuration for port/vlan AND vs. OR mirroring

EXHIBIT C

HP INVENTION DISCLOSURE

PDNO 200311029 DATE RCVD 3/24/03 ATTORNEY Amm

Instructions: The information contained in this document is **COMPANY CONFIDENTIAL** *and may not be disclosed to others without prior authorization. Submit this disclosure to the HP Legal Department as soon as possible. No patent protection is possible until a patent application is authorized, prepared, and submitted to the Government.*

Descriptive Title of Invention:
Remote Mirroring of Network Traffic

Name of Project: Back to the Future (BttF)

Product Name or Number:
None defined yet

Was a description of the invention published, or are you planning to publish? If so, the date(s) and publication(s):
Yes; planned for BttF product documentation, published near MR, expected Q1'05

Was a product including the invention announced, offered for sale, sold, or is such activity proposed? If so, the date(s) and location(s):
BttF products containing this invention are expected to MR worldwide in Q1'05

Was the invention disclosed to anyone outside of HP, or will such disclosure occur? If so, the date(s) and name(s):
No

If any of the above situations will occur within 3 months, call your IP attorney or the Legal Department now at 1-553-3061 or 408-553-3061.

Was the invention described in a lab book or other record? If so, please identify (lab book #, etc.)
Yes: Remote Port Monitoring.ppt, dated January 23, 2002, and
BttF Remote Mirroring Proposal - /nis/asic/bttf/ns/doc/bttf_remote_mirror_proposal.fm,
dated October 1, 2002 (attached)

Was the invention built or tested? If so, the date:
In progress (see MR date above)

Was this invention made under a government contract? If so, the agency and contract number:
No

Description of Invention: *Please preserve all records of the invention and attach additional pages for the following. Each additional page should be signed and dated by the inventor(s) and witness(es).*

A. Prior solutions and their disadvantages (if available, attach copies of product literature, technical articles, patents, etc.).
B. Problems solved by the invention.
C. Advantages of the invention over what has been done before.
D. Description of the construction and operation of the invention (include appropriate schematic, block, & timing diagrams; drawings; samples; graphs; flowcharts; computer listings; test results; etc.)

Signature of Inventor(s): Pursuant to my (our) employment agreement, I (we) submit this disclosure on this date: [3/7/03].

Employee No.	Name	Signature	Telnet	Mailstop	Entity & Lab Name
344862	Bruce LaVigne	Bruce LaVigne	785-4194	5672	PNB
368435	Paul Congdon	Paul J Congdon	785-5753	5662	PNB
464886	Mark Gooch	Mark Gooch	785-4077	5672	PNB

(If more than four inventors, include additional information on another copy of this form and attach to this document)

Signature of Witness(es): *(Please try to obtain the signature of the person(s) to whom invention was first disclosed.)*
The invention was first explained to, and understood by, me (us) on this date: [October 15, 2002]

Full Name	Signature	Date of Signature
Brian Ripley	Brian R. Ripley	3/7/03
Alan Albrecht	Alan Albrecht	3/7/03

Inventor & Home Address Information: *(If more than four inventors, include additional information on a copy of this form & attach to this document)*

Inventor's Full Name: Bruce LaVigne
Street: 1411 Voltaire Drive
City: Roseville — State: California — Zip: 95747
Do you have a Residential P.O. Address? P.O. BOX: NO — City: — State: — Zip:
Greeted as *(nickname, middle name, etc.)*: Bruce — Citizenship: United States

Inventor's Full Name: Paul Congdon
Street: 9489 Treelake Road
City: Granite Bay — State: California — Zip: 95746
Do you have a Residential P.O. Address? P.O. BOX: NO — City: — State: — Zip:
Greeted as *(nickname, middle name, etc.)*: Paul — Citizenship: United States

Inventor's Full Name: Mark Gooch
Street: 217 Hollow Oaks Court
City: Roseville — State: California — Zip: 95678
Do you have a Residential P.O. Address? P.O. BOX: NO — City: — State: — Zip:
Greeted as *(nickname, middle name, etc.)*: Mark — Citizenship: British

Inventor's Full Name:
Street:
City: — State: — Zip:
Do you have a Residential P.O. Address? P.O. BOX: — City: — State: — Zip:
Greeted as *(nickname, middle name, etc.)*: — Citizenship:

Remote Mirroring of Network Traffic

A. Prior solutions and their disadvantages

Current mirroring solutions copy packets to an additional port of the switch. The disadvantage is that the network analysis device must be directly attached to the switch which needs monitoring, limiting the usefulness of this technology.

In Cisco's RSPAN technology, the packet may be mirrored to a specific VLAN. This allows the analysis device to be on a different switch, but it still must be within the layer 2 domain of the traffic which is to be monitored, and in addition, the packet has now been modified (by adding/replacing the VLAN tag) from its original format. These two restrictions limit the usefulness of RSPAN.

Other network protocols may encapsulate packets, generally in software, to tunnel them across an arbitrary network (Generic Routing Encapsulation, GRE, for example). To date, this has not been coupled in hardware with mirroring traffic, plus even existing protocols such as GRE don't maintain complete layer 2 information such as the original VLAN tag, making exact replication of monitored traffic unattainable at a remote location.

Additionally, the choice of which packets to monitor is rudimentary in today's products. One can choose to mirror traffic which is ingress or egress on a port or a VLAN, that is all.

B. Problems solved by the invention

This remote mirroring technology allows exact copies of packets monitored from one switch to be encapsulated at line rate and transmitted through a general network and then decapsulated and sent out at a geographically remote site.

Also, the choice of which packets to monitor is enhanced by adding Access Control List (ACL) and hash lookups of the packets.

C. Advantages of the invention over what has been done before

As stated in the above two sections, today's solutions do not allow fully remote network troubleshooting, due to distance limitations, disruption of true packet format, lack of full media speed support, or a combination of these. This invention solves all of these problems, plus adds additional choices in how to select which packets to monitor as well.

D. Description of the construction and operation of the invention

See enclosed "BttF Remote Mirroring Proposal"